Document: **NIST CCRATWG 004 v2**

Date: January 25, 2011

Cloud Architecture Reference Models: A Survey

Cloud Computing is a new term for a long-held dream of computing as a utility that focuses on the delivery of scalable IT resources over the Internet as opposed to hosting and operating those resources locally. The NIST definition of cloud computing includes five essential characteristics (*on-demand self-service, broad network access, resource pooling, rapid elasticity, measured Service*), three service models (*cloud Infrastructure as a Service (IaaS*), *cloud Platform as a Service (PaaS), cloud Software as a Service (SaaS))*, and four deployment models (public cloud, private cloud, community cloud, hybrid cloud). In this research, we study the existent cloud architecture reference models.

A cloud architecture [11][12] is the structure of a cloud solution that uses Internet-accessible on-demand services. The cloud architecture comprises on-premise and cloud resources, services, middleware, software components, geo-location, the externally visible properties of those, and the relationships between them. The term also defines the structure of cloud services and cloud components, management, security, operation, and monitoring. Documentation is also necessary for a system's cloud computing architecture. Documenting facilitates communication between stakeholders, records early decisions about high-level design, and allows reuse of design components and patterns between projects.

A cloud architecture reference model [13] is an abstraction of cloud computing concepts and relationships that can be used to educate organizations. It can be used to create standards and guidelines to help apply those concepts. Groups such as the Distributed Management Task Force, IETF, Cloud Security Alliance, and Open Security Architecture are developing cloud reference models to help make informed decisions regarding how and whether to adopt these cloud technologies. Cloud providers (e.g. IBM, Cisco, etc.) and federal agencies (e.g. GSA, etc.) are also working on their own cloud reference models that meet their specific application requirements.

Cloud service providers, end users, and corporate IT departments are all expecting a cloud architecture reference model. The reference model is of great interest to service providers because it likely presents tremendous business opportunities for those who can successfully define and implement the new paradigm. Cloud end users are interested because services are reasonably priced and can be accessed from any browser giving access to the computing environment from any location and making collaboration much easier. Cloud service developers are interested because the model may deliver much faster development and implementation times, and promise to simplify the management of complex environments.

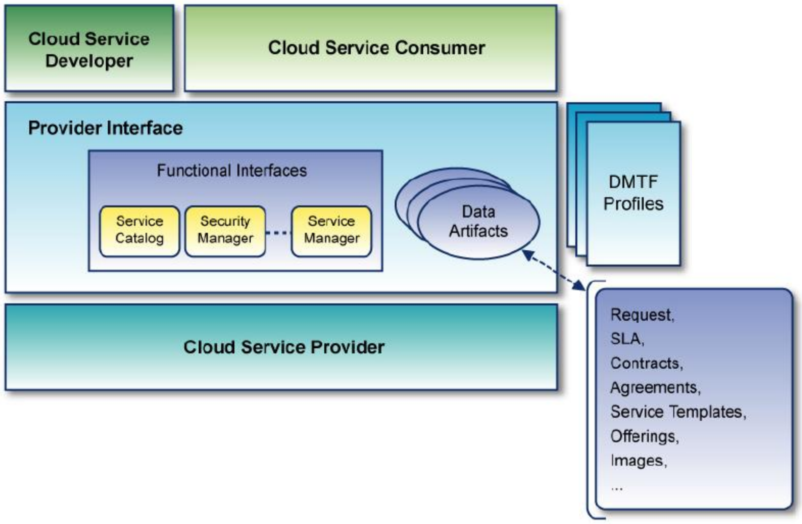
In this report, we survey the cloud reference architecture models proposed by known cloud organization, cloud providers, and federal agencies, which include:

- General reference models:
  - Distributed Management Task Force (DMTF): Cloud Service Reference Architecture
  - Cloud Computing Use Case Discussion Group: a taxonomy for cloud computing
  - IBM: Cloud Reference Architecture
  - Cloud Security Alliance: Cloud Reference Model
  - Cisco Cloud Reference Architecture Framework
  - IETF: Cloud Reference Framework

- Reference models focusing on specific application requirements:
  - Open Security Architecture: Secure Architecture Models
  - GSA: FCCI (Federal Cloud Computing Initiative)
  - Juniper Networks: Cloud-ready Data Center Reference Architecture
  - SNIA standard: Cloud Data Management Interface
  - Elastra: A Cloud Technology Reference Model for Enterprise Clouds

Finally, we conclude the report with a brief comparison of the presented models.

1. Distributed Management Task Force (DMTF): Cloud Service Reference Architecture [2]



DMTF's cloud Service Reference Architecture describes key components, such as actors, interfaces, data artifacts, and profiles and the interrelationships among these components.

1.1 Actors

The architecture has three primary actors: Cloud Service Provider, Cloud Service Consumer, and Cloud Service Developer. An organization may simultaneously play the roles of any combination of these actors.

- The Cloud Service Provider makes services available to Cloud Service Consumers at agreed service levels and costs. The services may be of any type or complexity. The Cloud Service Provider manages the technical infrastructure required for providing the services and provides billing and other reports to consumers.
- The Cloud Service Consumer represents an organization or individual who contracts for services with Cloud Service Providers and then uses those services. The Cloud Service Consumer could be another Cloud Service Provider. The Cloud Service Consumer is responsible for selecting the appropriate services, arranging payment for the services, and performing the administration necessary to use those services, such as managing user identities.
- The Cloud Service Developer designs and implements the components of a service. The Cloud Service Developer describes the service in a service template. The Cloud Service Developer interacts with the Cloud Service Provider to deploy the service components based on the description in the templates that the Cloud Service Provider may customize before making them available as service offerings.

1.2 Interfaces and Data Artifacts

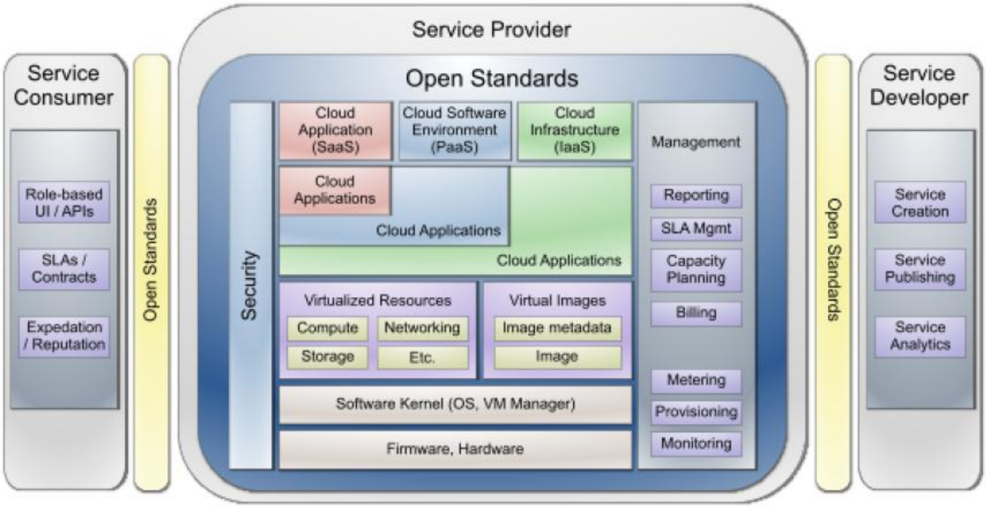
A provider interface defines how the developer and consumer interact with the provider. This architecture differentiates between service endpoints that accept (and respond to) messages over a protocol based on some message exchange pattern (functional interfaces) and the data elements and operations that an interface can support (data artifacts). The interface comprises both functional interfaces and data artifacts.

- Functional interfaces are programming interfaces (for example, APIs). Through these interfaces, Cloud Service Developers and Cloud Service Consumers interact with providers to request, deploy, administer, and use services. Examples of likely functional interfaces are:
  - A Service Catalog, through which service offerings are offered, requested, and managed;
  - A Security Manager, through which the security-related aspects of a cloud are managed;
  - A Service Manager, through which instances of deployed services are managed and modified.
- Data artifacts are exchanged over the functional interfaces. In this context, a data artifact definition describes the semantic content and the specific format (for example, the XML schema definition that describes the XML payload). Examples of data artifact types include service requests, service level-agreements (SLAs) and other contracts, service templates, service offerings, and images that contain applications. For example, a customizable contract template that includes the customer request, SLA, and security requirements is needed to support the service catalog interface. SLA, security requirements, and resource specifications are used to build offerings.

1.3 DMTF Profiles

DMTF profiles are normative specializations or extensions of the interfaces and artifacts, or combinations of them, which are useful in addressing certain contexts, such as those of interest to a security manager or a contract billing administrator. Profiles may be used to simplify the interactions and the potentially complex definitions and negotiation needed to request, manage, and use services. A profile may also specify the use of particular standards that are useful in the profile's target environment and use cases. A profile represents a view into the provider interface.

2. Cloud Computing Use Case Discussion Group: a taxonomy for cloud computing [1]



In this diagram, Service Consumers use the services provided through the cloud, Service Providers manage the cloud infrastructure and Service Developers create the services themselves.

2.1 Service Consumer

The Service Consumer is the end user or enterprise that actually uses the service, whether it is Software, Platform, or Infrastructure as a Service.

Depending on the type of service and their role, the Service Consumer works with different user interfaces and programming interfaces. Some user interfaces look like any other application; the Service Consumer does not need to know about cloud computing as they use the application. Other user interfaces provide administrative functions such as starting and stopping virtual machines or managing cloud storage. Service Consumers writing application code use different programming interfaces depending on the application they are writing.

Service Consumers work with SLAs and contracts as well. Typically these are negotiated via human intervention between the Service Consumer and the Service Provider. The expectations of the Service Consumer and the reputation of the Service Provider are a key part of those negotiations.

2.2 Service Provider

The Service Provider delivers the service to the Service Consumer. The actual task of the provider varies depending on the type of service:

- For Software as a Service, the Service Provider installs, manages, and maintains the software. The Service Provider does not necessarily own the physical infrastructure in which the software is running. Regardless, the Service Consumer does not have access to the infrastructure; they can access only the application.
- For Platform as a Service, the Service Provider manages the cloud infrastructure for the platform, typically a framework for a particular type of application. The Service Consumer’s application cannot access the infrastructure underneath the platform.
- For Infrastructure as a Service, the Service Provider maintains the storage, database, message queue, other middleware, or the hosting environment for virtual machines. The Service Consumer uses that service as if it were a disk drive, database, message queue, or machine, but cannot access the infrastructure that hosts it.

In the Service Provider diagram, the lowest layer of the stack is the firmware and hardware on which everything else is based. Above that is the software kernel, either the operating system or virtual machine manager, which hosts the infrastructure beneath the cloud. The virtualized resources and images include the basic cloud computing services such as processing power, storage, and middleware. The virtual images controlled by the VM manager include both the images themselves and the metadata required to manage them.

Crucial to the Service Provider’s operations is the management layer. At a low level, management requires metering to determine who uses the services and to what extent, provisioning to determine how resources are allocated to consumers, and monitoring to track the status of the system and its resources.

At a higher level, management involves billing to recover costs, capacity planning to ensure that consumer demands will be met, SLA management to ensure that the terms of service agreed to by the Service Provider and the Service Consumer are adhered to, and reporting for administrators.

Security applies to all aspects of the Service Provider's operations (the many levels of security requirements are beyond the scope of this paper). Open standards apply to the Service Provider's operations as well. A well-rounded set of standards simplify operations within the Service Provider and interoperability with other Service Providers.

2.3 Service Developer

The Service Developer creates, publishes and monitors the cloud service. These are typically "line-of-business" applications that are delivered directly to end users via the SaaS model. Applications written at the IaaS and PaaS levels will subsequently be used by SaaS Service Developers and Service Providers.

Development environments for service creation vary. If Service Developers are creating a SaaS application, they are most likely writing code for an environment hosted by a Service Provider. In this case, publishing the service is deploying it to the Service Provider's infrastructure.

During service creation, analytics involve remote debugging to test the service before it is published to Service Consumers. Once the service is published, analytics allow Service Devlopers to monitor the performance of their service and make changes as necessary.

3. IBM Cloud Reference Architecture [14]~[20]

3.1 Overview

The next picture shows a typical cloud management architecture overview. It consists of three major building blocks based on the high-level roles that are involved in cloud computing: Cloud Service Consumer, Cloud Service Provider, and Cloud Service Developer.

3.1.1 Cloud Service Consumer

The Cloud Service Consumer includes every user and system that uses or consumes resources of the cloud.

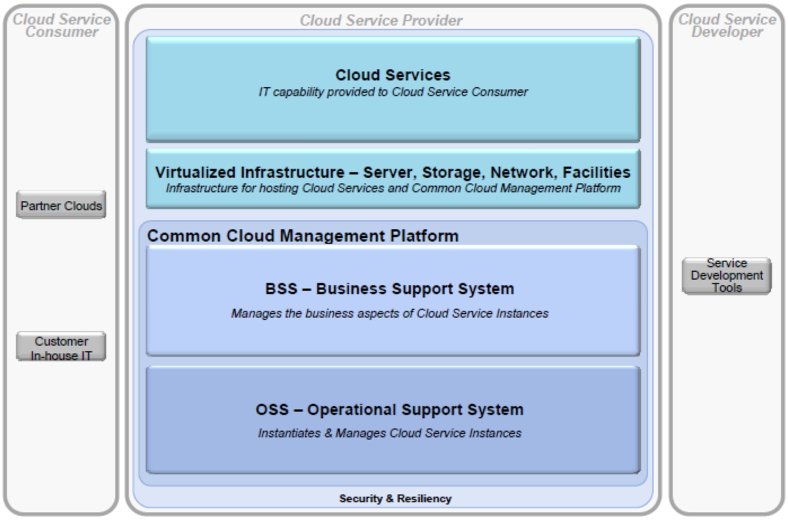
The functions and services offered by cloud computing start with the needs of the Cloud Service Consumer, who makes a request for services and resources through a Self-Service portal. Cloud applications then search for resources to match the request using a portfolio of cloud services. Access is provided back to the Cloud Service Consumer through the portal.

3.1.2 Cloud Service Provider

The Cloud Service Provider offers cloud services, supplies Infrastructure, and a Common Cloud Management Platform. The Common Cloud Management Platform (CCMP) consists of two major building blocks:

- The Business Support Services (BSS), consisting of the business-relevant platform services;
- The Operational Support Services (OSS), representing the more infrastructure-related and operational aspects of the Common Cloud Management Platform.

IBM Cloud Reference Architecture


Cloud Service Consumer
Partner Clouds
Customer In-house IT
Cloud Service Provider
Cloud Services
IT capability provided to Cloud Service Consumer
Virtualized Infrastructure – Server, Storage, Network, Facilities
Infrastructure for hosting Cloud Services and Common Cloud Management Platform
Common Cloud Management Platform
BSS – Business Support System
Manages the business aspects of Cloud Service Instances
OSS – Operational Support System
Instantiates & Manages Cloud Service Instances
Security & Resiliency
Cloud Service Developer
Service Development Tools

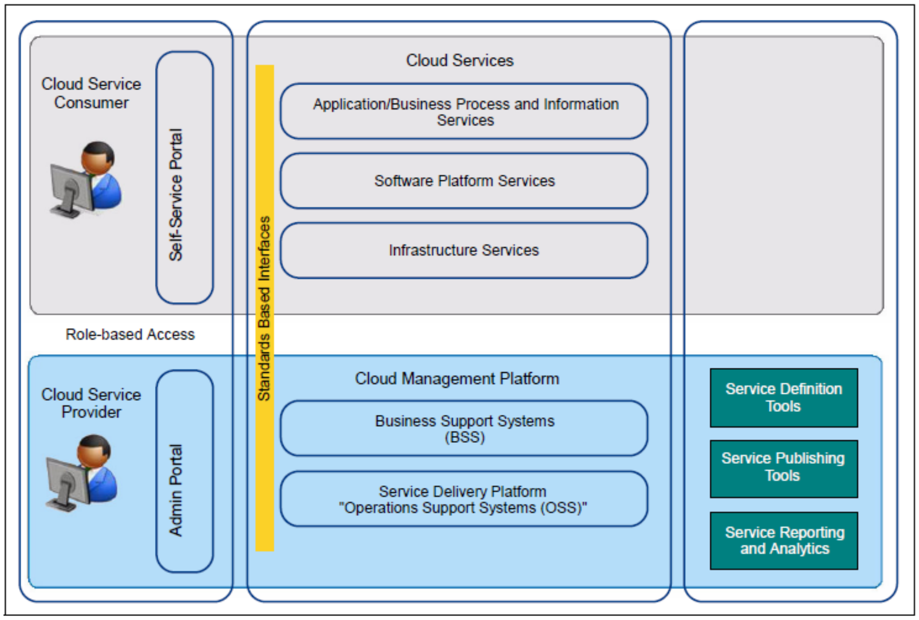
The Cloud Service Provider role includes the following:

- Service operations managers: Manage technical infrastructure required for providing cloud services;
- Service business managers: Offer all types of services created by the Cloud Service developer;
- Service transition managers: Responsible for enabling a client to use the cloud service, including on-boarding, integration, and process adoption.

The Cloud Service Provider performs the following tasks:

- Build services by (optionally) consuming services provided by other Cloud Service Providers.
- Offer services based on a management infrastructure.
- Host services created by other service creators (on top of their own services).

Cloud services are made available to the consumer by the administrator. The administrator uses the Admin Portal to publish the services that are offered as part of the cloud, as well as to prepare resources for use within the cloud. This portal is the interface to a number of tools and reports as well, as shown in next figure.


Cloud Service Consumer
Self-Service Portal
Role-based Access
Cloud Service Provider
Admin Portal
Standards Based Interfaces
Cloud Services
Application/Business Process and Information Services
Software Platform Services
Infrastructure Services
Cloud Management Platform
Business Support Systems (BSS)
Service Delivery Platform "Operations Support Systems (OSS)"
Service Definition Tools
Service Publishing Tools
Service Reporting and Analytics

3.1.2.1 Admin Portal

The Admin Portal provides administrators with the capability to add, manage, support, and administer private test cloud services and resources in order to fulfill Cloud Service Consumer requests. The functions of the Admin Portal are to:

- Manage growth, outages, changes, and other life cycle aspects of the cloud;
- Automate actions in the cloud based on monitoring metrics and threshold measurements;
- Control the process workflow to be documented, record approvals, and measure key performance indicators for SLA adherence.

3.1.3 Cloud Service Developers

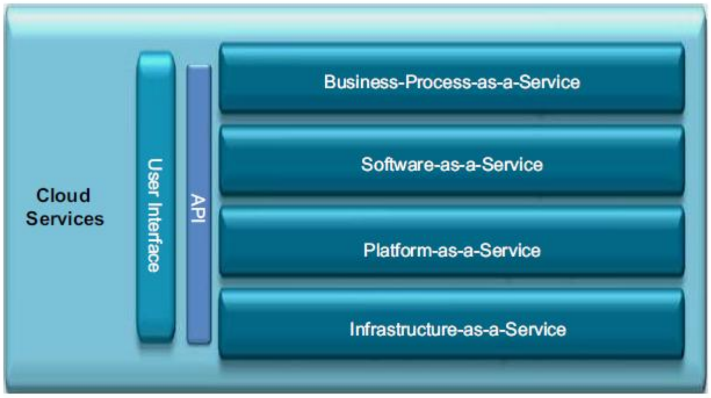
A Cloud Service Developer uses Service Development Tools to develop new cloud services, which includes both the development of runtime artifacts (for example, database persistence, transactional handling, etc.) and management-related aspects (for example, monitoring, metering, provisioning, etc.). In this context, the service development tools support the Cloud Service Developer in creating a service template and a service offering, where the service template defines how the Common Cloud Management Platform (CCMP) OSS functionality is used in the context of the respective cloud service and the service offering specifies how the CCMP BSS functionality is used in the context of the respective cloud service.

In the context of a particular IaaS or PaaS offering, there might also be tooling to develop artifacts that are specific to the particular cloud service.

In summary, there are two categories of service development tooling: tooling to develop a cloud service by itself and tooling to develop artifacts that are specific to that cloud service.

3.2 Cloud Services

Cloud Services represent any type of (IT) capability that is provided by the Cloud Service Provider to Cloud Service Consumers. Typical categories of Cloud Services are infrastructure, platform, software, or Business Process Services. In contrast to traditional (IT) services, Cloud Services have attributes associated with cloud computing, such as a pay-per-use model, self-service usage, flexible scaling, and shared underlying IT resources.



3.2.1 Business Process as a Service

Business Process as a Services are focused on providing existing business processes through a cloud. If there is an existing process with steps that are known it can be provided as a service within the catalog. This allows the Cloud Service Provider to automate any steps within the process while leaving the changes transparent to the Cloud Service Consumer.

3.2.2 Software as a Service

Software Services allow a Cloud Service Consumer to select a specific software instance that they want created without the need to be aware of where and how it will be hosted. For example, a developer can request a new database instance without having to be aware of what OS or hardware the database will run on. This allows the Cloud Service Consumer to focus on the characteristics of the application and gives the Cloud Service Provider the freedom to fulfill the request with any resources that will meet the need.

3.2.3 Platform as a Service

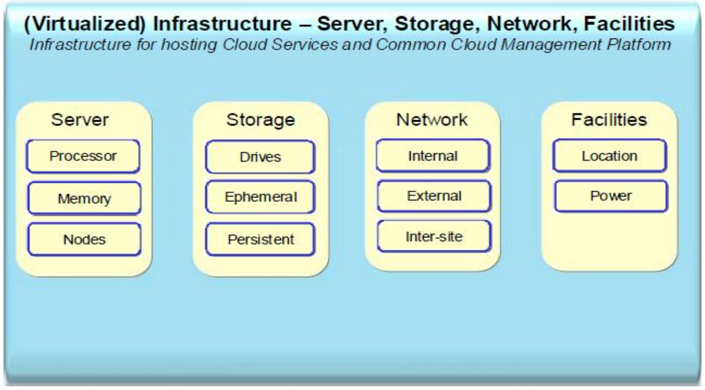
Platform Service offerings include workflow facilities for design, development, testing, deployment, and hosting, as well as services that enable team collaboration, web service integration and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation, and developer community facilitation. These services are provisioned as an integrated solution over the Web.

3.2.4 Infrastructure Services

Infrastructure Services allow for the provisioning of standardized computing resources. They allow a consumer to request and receive a new computer instance without needing to focus on IT concerns such as network placement and hardware availability.

3.3 Virtualized Infrastructure

The Virtualized Infrastructure includes all infrastructure elements needed on the Cloud Service Provider side to provide cloud services, which include facilities, servers, storage, and network resources and how these resources are wired up, placed within a data center, etc. In case of virtualization, this also includes virtualization information, such as hypervisors. It does not include any virtualization management software because that is part of the virtualization management component of the "Operational Support Services".


(Virtualized) Infrastructure – Server, Storage, Network, Facilities
Infrastructure for hosting Cloud Services and Common Cloud Management Platform
Server
Processor
Memory
Nodes
Storage
Drives
Ephemeral
Persistent
Network
Internal
External
Inter-site
Facilities
Location
Power

3.4 Cloud Management Platform

The cloud management platform is the set of tools and capabilities that provide services to the Cloud Service Consumer. Most of the previous items were about services that were consumed. Cloud management is about providing resources to the system so that services can be consumed.

The Common Cloud Management Platform (CCMP) contains a set of business and operational management focused services that must be used by Cloud Services to actually be a cloud service. The CCMP is responsible for:

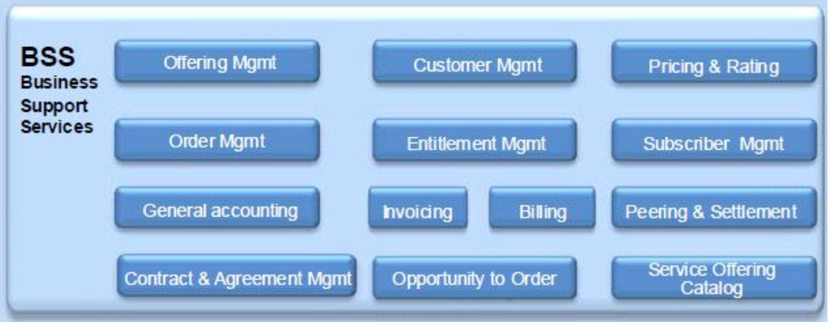
- Delivering instances of Cloud Services of any category to Cloud Service Consumers
- The ongoing management of all Cloud Service instances from a Cloud Service Provider perspective
- Allowing Cloud Service Consumers to manage their Cloud Service instances in a self-service fashion

The CCMP is split into two main elements: Operational Support Services, and Business Support Services.

3.4.1 Business Support Systems

Business Support Systems (BSS) represent the set of business-related services exposed by the CCMP dealing with clients, supporting processes such as taking orders, processing bills, and collecting payments. It includes the components used to run business operations that are client-facing.

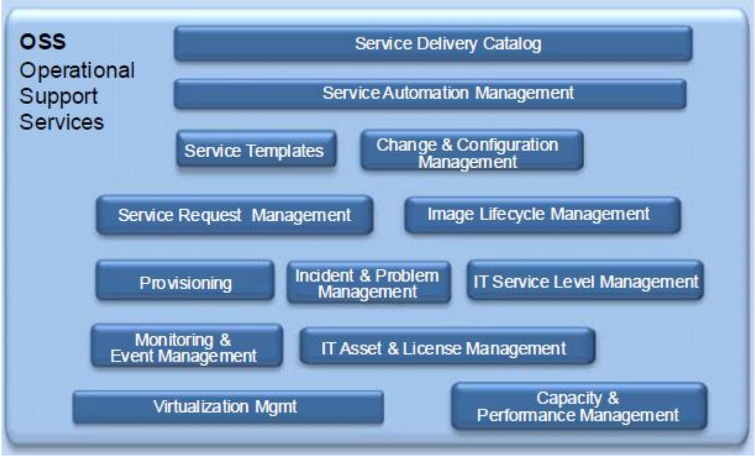
The BSS provides services that either enable the Cloud Service Provider or facilitates certain task to deliver the cloud from a business perspective. It contains the services offering management, customer management, pricing and rating, order management, entitlement management, subscriber management, general accounting, invoicing, billing, peering and settlement, contract and agreement management, opportunity to order, metering, analytics and reporting, and the service offer catalog.



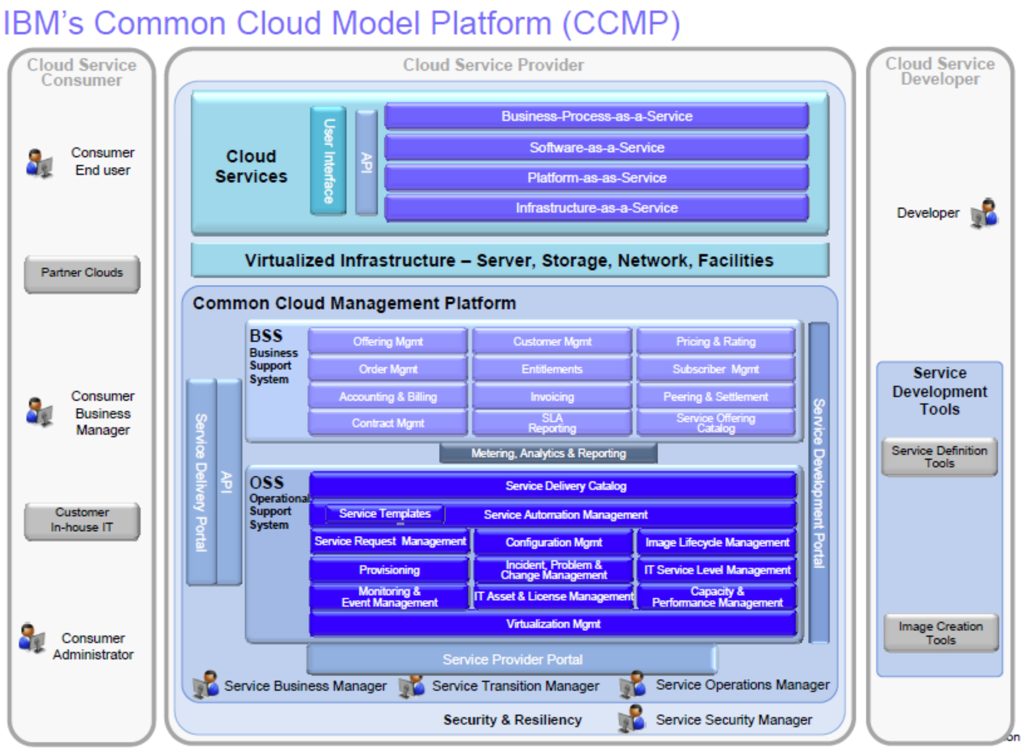
3.4.2 Operational Support Services

OSS represents the set of operational management and technical-related services exposed by the CCMP, which must be exploited by Cloud Service Developers to take advantage of the common cloud management platform.

The OSS contains the following services: service delivery catalog, service template, service automation management, service request management, change and configuration management, image life cycle management, provisioning, incident and problem management, IT service level management, monitoring and event management, IT asset and license management, capacity and performance management, and virtualization management.


OSS
Operational Support Services
Service Delivery Catalog
Service Automation Management
Service Templates
Change & Configuration Management
Service Request Management
Image Lifecycle Management
Provisioning
Incident & Problem Management
IT Service Level Management
Monitoring & Event Management
IT Asset & License Management
Virtualization Mgmt
Capacity & Performance Management

In summary, IBM cloud reference architecture can be seen in the following figure:



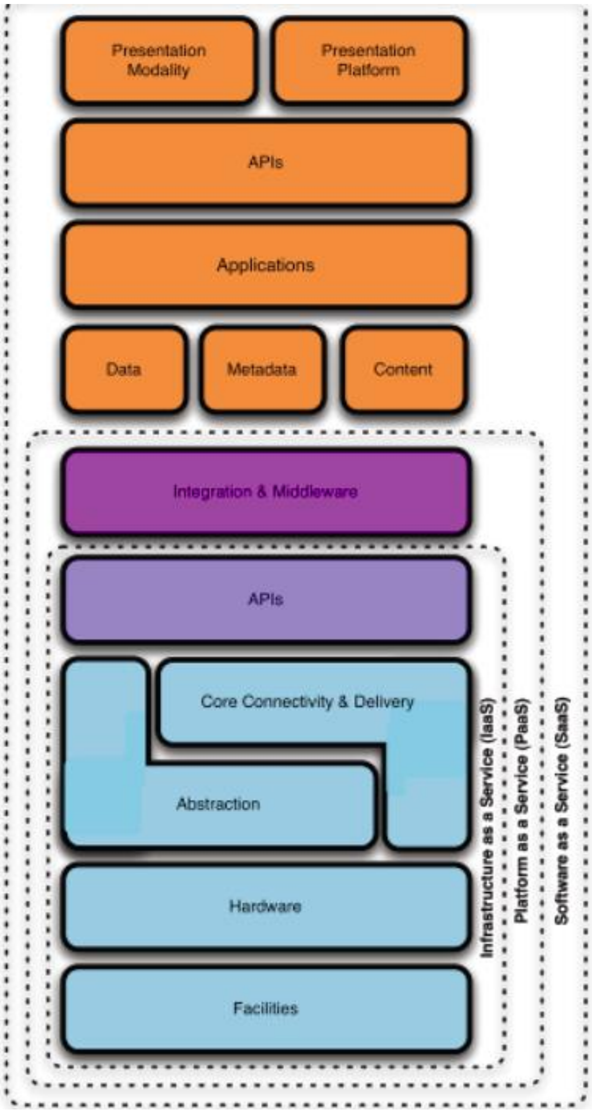
4. Cloud Security Alliance: Cloud Reference Model [4]

Understanding the relationships and dependencies between Cloud Computing models is critical to understanding Cloud Computing security risks. IaaS is the foundation of all cloud services, with PaaS building upon IaaS and SaaS in turn building upon PaaS, as described in the Cloud Reference Model diagram. In this way, just as capabilities are inherited so are information security issues and risk. It is important to note that commercial cloud providers may not neatly fit into the layered service models. Nevertheless, the reference model is important for relating real-world services to an architectural framework and understanding the resources and services requiring security analysis.

IaaS includes the entire infrastructure resource stack from the facilities to the hardware platforms that reside in them. It incorporates the capability to abstract resources (or not), as well as deliver physical and logical connectivity to those resources. Ultimately, IaaS provides a set of APIs which allow management and other forms of interaction with the infrastructure by consumers.

PaaS sits atop IaaS and adds an additional layer of integration with application development frameworks, middleware capabilities, and functions such as database, messaging, and queuing that allow developers to build applications upon the platform and whose programming languages and tools are supported by the stack.

SaaS is built upon the underlying IaaS and PaaS stacks and provides a self-contained operating environment used to deliver the entire user experience including content, presentation, application(s), and management capabilities.



It should be clear that there are significant trade-offs to each model in terms of integrated features, complexity vs. openness (extensibility), and security.

- Generally, SaaS provides the most integrated functionality built directly into the offering with the least consumer extensibility, and a relatively high level of integrated security (at least, the Cloud Service Provider bears a responsibility for security).
- PaaS is intended to enable developers to build their own applications on top of the platform. As a result it tends to be more extensible than SaaS at the expense of customer-ready features. This tradeoff extends to security features and capabilities, where the built-in capabilities are less complete but have more flexibility to layer on additional security.
- IaaS provides few, if any, application-like features, but allows enormous extensibility. This generally means fewer integrated security capability and functionality beyond protecting the

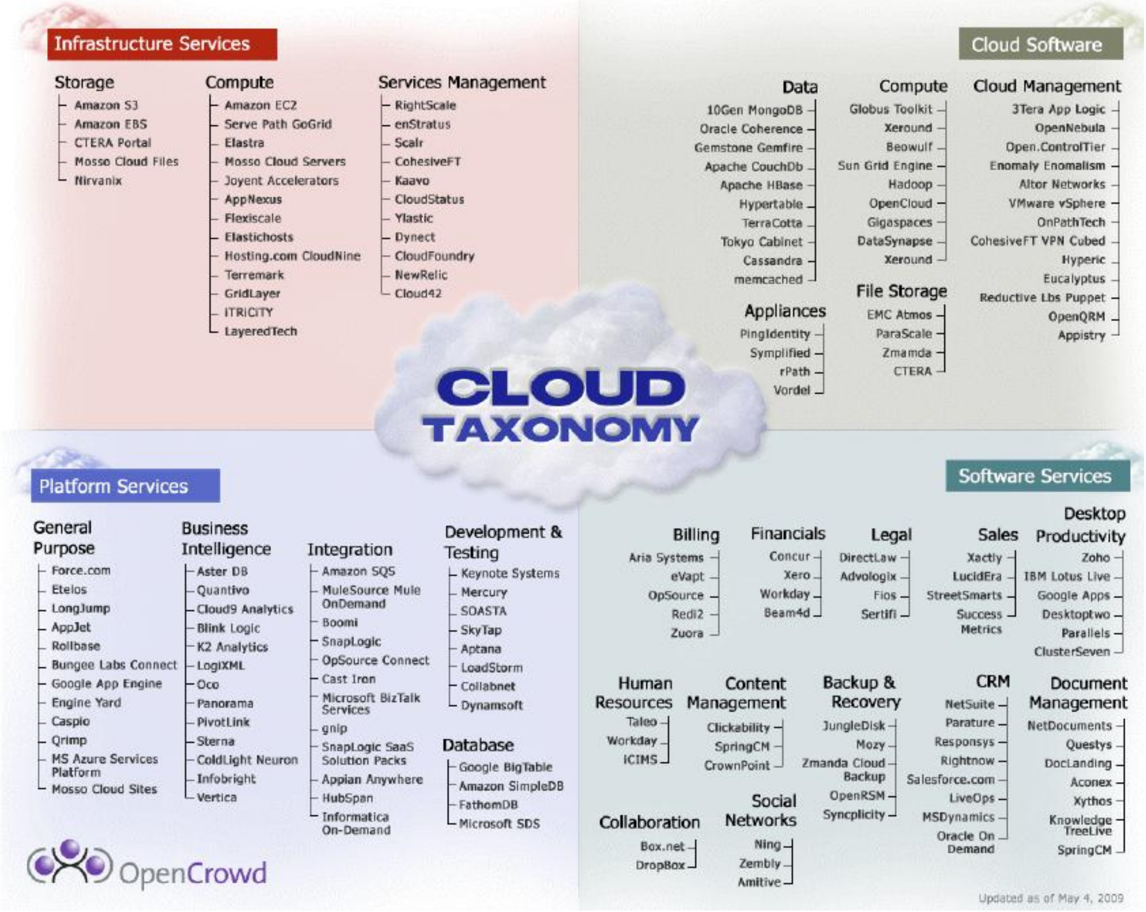
infrastructure itself. This model requires that operating systems, applications, and content be managed and secured by the Cloud Service Consumer.

The key takeaway for security architecture is that the lower down the stack the Cloud Service Provider stops the more security capabilities and management Cloud Service Consumers are responsible for implementing and managing themselves.

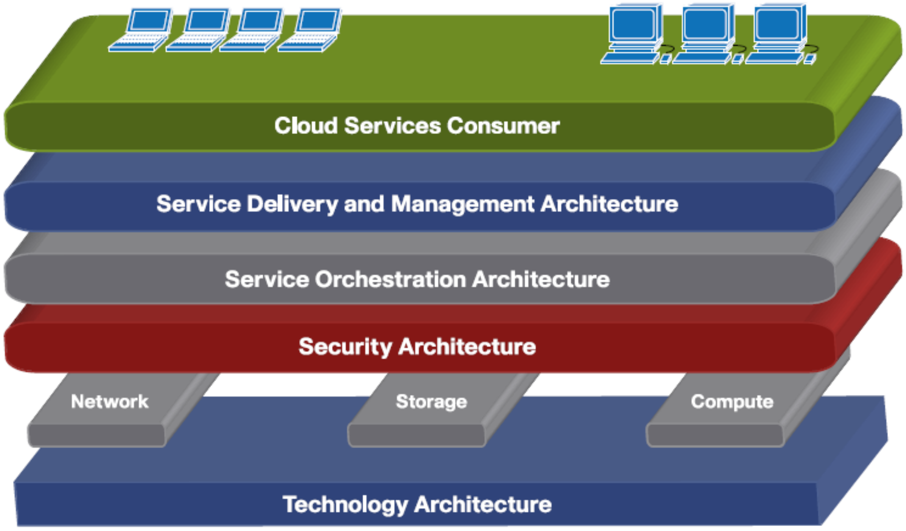
In the case of SaaS, service levels, security, governance, compliance, and liability expectations of the service and Cloud Service Provider are contractually stipulated, managed to, and enforced. In the case of PaaS or IaaS it is the responsibility of the Cloud Service Consumer's system administrators to effectively manage the same, with some expecation that the Cloud Service Provider will be responsible for securing the underlying platform and infrastructure components to ensure basic service availability and security. It should be clear in either case that one can assign/transfer responsibility but not necessarily accountability.

Narrowing the scope or specific capabilities and functionality within each of the cloud delivery models or employing the functional coupling of services and capabilities across them may yield derivative classifications. For example, "Storage as a Service" is a specific sub-offering within the IaaS "family".

The OpenCrowd taxonomy shown below demonstrates the growing number of solutions available today across each of the previously defined models.



5. Cisco Cloud Reference Architecture Framework [21]



Cisco has developed a cloud reference architecture model that portrays the architectural layers connected via APIs and repositories. If we study the framework more closely, the following aspects can be articulated.

- Technology Architecture: This consists of three salient domains: network, computing, and storage. This layer hosts all the services that are delivered to a Cloud Service Consumer.
- Security layer: The key takeaway in this layer is that security is blanketed as an end-to-end architecture across all aspects of the framework. Security is considered one of the key challenges to be solved in a cloud framework; hence, it has to be accounted for in a comprehensive sense.
- Service Orchestration layer: This is immplemented with configuration repository enablers. The configuration repository stores key information such as the service catalogue, an asset inventory, and resource-to-service mapping. This layer is an important layer because it maps the technology components to the service components and serves as a reference point during service provisioning. The service orchestration layer is the “glue” that integrates the lower layers to create a service for delivery.
- Service Delivery and Management layer: This is the layer where the infrastructure and service management functions take place.
- Cloud Services Consumers layer: The Cloud Service Consumer-facing layer, usually exposed via a portal-like solution. This is the layer where service is defined, requested, and managed by the Cloud Service Consumer.

Let’s walk through a use case scenario where this framework is utilized.

1. The Cloud Service Consumer logs on to a cloud portal and verifies/updates credentials and information.
2. Based on the consumer entitlement, a selected set of services are identified and presented for definition.

3. The end user selects the service for consumptions and triggers a service-provisioning request.
4. Resources are marked as reserved for service, and a new request is created for service provisioning.
5. The individual domains of compute, network, and storage are configured and provisioned with requested security and service-level agreements (SLAs), for service delivery.

Hence, this framework provides a working structure to create, define, orchestrate, and deliver IT service via a cloud.

6. IETF: Cloud Reference Framework

IETF published its internet-draft on intra-cloud and inter-cloud reference frameworks on Dec.31, 2010.

This reference framework documents basic functions or layers to support the general requirements of Cloud Applications and Services. This reference framework can be used to standardize the interfaces between the functions or layers.

Basically, the Cloud Framework can be divided into:

- Four horizontal layers:
  - Application/Service Layer(ASL)
  - Resource Control Layer(RCL)
  - Resource Abstract and Virtualization Layer(RAVL)
  - Physical Resource Layer(PRL)
- One stacked vertical layer to support
  - Configuration management, registry, logging and auditing, security management, and service level agreement (SLA) management

6.1 Application/Service Layer

The Application/Service layer defines the requirements of the basic functional entities based on the virtual resources needed to perform any tasks. The tasks are classified according to the 3 services models: IaaS, PaaS, and SaaS. Some cloud services are illustrated as an example of applications, such as:

- Server, desktop, database and VLAN for IaaS;
- Development environment and test environment for PaaS;
- Business, consumer, network and communication applications for SaaS.

6.2 Resources Control Layer

The Resources Control layer manages virtual resources, ensuring that the resources are efficient, secure and reliable. With the interface of virtual resources, the layer integrates the resources as a whole supplied to the layer above. The layer has the following responsibilities:

- Resource security management. Resources must be accessed and owned by the appropriate user. There are several function modules to fulfill this responsibility, including resource admission control, resource authentication, and authorization control;

```
             +------------------------------+                               +-----------------+
             |         Cloud Portal         |                               |                 |
             |      (Public & Private)      |                               |                 |
             +------------------------------+                               |                 |
                            |                                               |                 |
                            |                                               |                 |
+----------------------------------------------------------------------+    |                 |
|                            Application/Service Layer                 |    |      Cloud      |
| +-----------+ +------+ +-------------------------------------------+ |    |    Management   |
| |           | |      | |   SaaS(Applications)                      | |    |                 |
| |           | |      | | +----------------+ +--------------------+ | |    |                 |
| |           | |      | | |  BusinessApps  | |    ConsumerApps    | | |    |                 |
| |           | |      | | |(Mobile payment)| |(Mobile Data backup)| | |    |                 |
| | +-------+ | |      | | +----------------+ +--------------------+ | |    |                 |
| | |Desktop| | |      | |   +------------+   +--------------------+ | |    |                 |
| | +-------+ | |      | |   |NetworkApps |   | CommunicationApps  | | |    |                 |
| |           | |      | |   |(Hosted PBX)|   |(VoIP,Video Service)| | |    | +-------------+ |
| |           | |      | |   +------------+   +--------------------+ | |    | |Configuration| |
| |           | |      | +-------------------------------------------+ |    | |  Management | |
| |           | |      +---------------------------------------------+ |    | +-------------+ |
| | +-------+ | |  PaaS(Software Environment)                        | |    |                 |
| | |Server | | |              +-----------+  +-----------+          | |<-->| +-------------+ |
| | +-------+ | |              |Development|  |Test       |          | |    | | Registry &  | |
| |           | |              |Environment|  |Environment|          | |    | | Repository  | |
| |           | |              +-----------+  +-----------+          | |    | +-------------+ |
| |           | +----------------------------------------------------+ |    |                 |
| |           +------------------------------------------------------+ |    | +-------------+ |
| | IaaS(Infrastructure)        +----------+  +--------+             | |    | |   Audit &   | |
| |                             | Database |  |Security|             | |    | |   Logging   | |
| |                             +----------+  +--------+             | |    | +-------------+ |
| |                             +----------+  +--------+             | |    |                 |
| |                             |MiddleWare|  |  VLAN  |             | |    | +-------------+ |
| |                             +----------+  +--------+             | |    | |     SLA     | |
| +------------------------------------------------------------------+ |    | +-------------+ |
+----------------------------------------------------------------------+    |                 |
         |                      |                        |                  | +-------------+ |
         |                      |                        |                  | |   Security  | |
+----------------------------------------------------------------------+    | +-------------+ |
|                               Resource Control Layer                 |    |                 |
| +---------+ +--------------+ +--------+ +------------+ +-----------+ |    |                 |
| |Resource | |Resource      | |Resource| |Resource    | |Inter-Cloud| |    |                 |
| |Admission| |Authentication| |Schedule| |Availability| |Resource   | |<-->|                 |
| |Control  | |&Authorization| |Control | |Control     | |Control    | |    |                 |
| |         | |Control       | |        | |            | |           | |    |                 |
| +---------+ +--------------+ +--------+ +------------+ +-----------+ |    |                 |
+----------------------------------------------------------------------+    |                 |
         |                      |                        |                  |                 |
         |                      |                        |                  |                 |
+----------------------------------------------------------------------+    |                 |
|                  Resource Abstract&Virtualization Layer              |    |                 |
| +------------------------------------------------------------------+ |    |                 |
| |                                        Virtualized Resource      | |    |                 |
| | +---------+  +--------+  +------+  +-----------------+  +------+ | |    |                 |
| | |   V-    |  |  V-    |  |  V-  |  |       V-        |  | VPN  | | |    |                 |
| | |Computing|  |Storage |  |Switch|  |Network Interface|  +------+ | |    |                 |
| | +---------+  +--------+  +------+  +-----------------+           | |    |                 |
| | +---------+  +--------+  +------+  +-----------------+           | |    |                 |
| | |   V-    |  |   V-   |  |  V-  |  |       V-        |  +------+ | |    |                 |
| | |Database |  |FireWall|  |Router|  |  Network Link   |  |Other | | |    |                 |
| | +---------+  +--------+  +------+  +-----------------+  +------+ | |    |                 |
| +------------------------------------------------------------------+ |<-->|                 |
|----------------------------------------------------------------------|    |                 |
|      +----+       +----+       +----+         +----+       +----+    |    |                 |
|      | VM |       | VM |       | VM |         | VM |       | VM |    |    |                 |
| +------------------------------------------------------------------+ |    |                 |
| |                             Hypervisor                           | |    |                 |
| +------------------------------------------------------------------+ |    |                 |
+----------------------------------------------------------------------+    |                 |
         |                      |                        |                  |                 |
         |                      |                        |                  |                 |
+----------------------------------------------------------------------+    |                 |
|                               Physical Resource Layer                |    |                 |
| +----------+ +-------------+ +-------------------------------------+ |    |                 |
| |  SERVER  | |  STORAGE    | |          NETWORK                    | |<-->|                 |
| | +------+ | | +---------+ | | +------+ +--------+ +------+        | |    |                 |
| | | CPU  | | | |Hard Disk| | | |Router| |FireWall| |Switch|        | |    |                 |
| | +------+ | | +---------+ | | +------+ +--------+ +------+        | |    |                 |
| | +------+ | |             | | +-----------------+ +------------+  | |    |                 |
| | |MEMORY| | |             | | |Network Interface| |Network Link|  | |    |                 |
| | +------+ | |             | | +-----------------+ +------------+  | |    |                 |
| +----------+ +-------------+ +-------------------------------------+ |    |                 |
+----------------------------------------------------------------------+    +-----------------+
```

- Resource schedule control. The layer manages resources in the form of a resource pool. In a resource pool, the layer balances the virtual resources on physical equipment to achieve higher hardware utilization. Virtual resources can be migrated between physical equipment if necessary and also can be allocated according to user's priority.
- Inter-cloud resource control. Resources in a cloud can be shared with another cloud in some circumstances, so a cloud must control resources in the other cloud and supply cloud services to end users. End users have no need to know where the resources are from.
- Resource availability control. The layer supports fault-tolerance on resources. It can allocate another copy of resources as a backup, and switch over when some faults raised.

6.3. Resource Abstraction and Virtualization Layer

Physical resources at the lowest level are the most complex to share among multiple users. There are several hardware details that don't need to be visible to users, so we need a level of abstraction. In fact, these physical resources are abstracted first. The function of the Resource Abstraction and Virtualization layer is to convert physical resources to virtual resources. Virtual resources are contained in a resource pool. Resources can be allocated to users from the resource pool and released back into the resource pool when they are no longer needed.

6.4. Physical Resource Layer

The Physical Resource Layer includes:

- CPU
- Memory
- Hard Disk
- Network Interface Card
- Network Link
  - Ports
  - Bandwidth

6.5. Cloud Management Layer

The Cloud Management Layer (CML) provides monitoring and administration of the cloud network platform to keep the whole cloud operating normally.

Key features of the Cloud Management Layer include:

- Automatically deploying the cloud system based on the configuration data and policy;
- Real-time monitoring and alerting of cloud status, resource usage, and performance of cloud;
- Reporting and charting of historical events and performance metrics;
- Flexible IT management and operational status displays;
- Authenticating/Authorizing the published cloud service registry;
- Auditing the cloud environment to check whether its running smoothly;
- Controlling the SLA implemented in the cloud system;
- Maintenance concerned with performing repairs, upgrades, and joining new nodes into the Cloud;
- Providing a security mechanism for the Cloud.

Basically, CML includes four Functions:

- Cloud Configuration Management: Cloud Configuration Management (CCM) is responsible for establishing and maintaining the consistent performance of the cloud system or product and its functional and physical attributes throughout its life-cycle. It mainly focuses on configuring the cloud system and retrieving the configuration information automatically.
- Cloud Service Registry and Audit Management:
  - The Service Registry/Repository provides management and governance capabilities that enable the published cloud service to be authenticated in the cloud system and accessed by service client. It facilitates storing, accessing and managing service information, called service metadata, so that the cloud service can be easily published, selected, invoked, enriched, governed, and reused.
  - Cloud Audit Management (CAM) provides an agent through which cloud providers and authorized consumers automate the Audit, Assertion, Assessment, and Assurance of the cloud infrastructure (IaaS), platform (PaaS), and application (SaaS) environments to reduce the risk. A common interface and namespace can be used by the CAM to facilitate these audit functions.
- Cloud SLA Management: Cloud SLA Management (CSM) is used to control the usage and receipt of resources from and by third parties. The strategy of CSM includes the negotiation of the contract and the monitoring of its realization in real-time. Thus, CSM encompasses the SLA contract definition (basic schema within QoS parameters), the SLA negotiation, the SLA monitoring, and the SLA enforcement. CSM must also define rate reductions and discounts that are applied if a service provider fails to meet the desired service parameters or does not fulfill an agreement.
- Cloud Service Security Management: Cloud Service Security (CSS) provides a set of mechanisms (e.g. IP address filtering , message integrity and confidentiality, private key encryption, dynamic session key encryption, user authentication, and service certification) to protect Cloud Services and their operating environment from damage.

6.6 Inter-Cloud Framework

```
                          Inter-Cloud Interface
                                   |
+---------------------------+   ICI-1      +-------------------------+
|             Cloud 1       |<-----|------>|   Cloud 2               |
|            +-----------+  |      |       |  +-----------+          |
|            |Inter-Cloud|  |   ICI-2      |  |Inter-Cloud|          |
|            |Resource   |  |<-----|------>|  |Resource   |          |
|            |Control    |  |      |       |  |Control    |          |
|            +-----------+  |   ICI-3      |  +-----------+          |
|                           |<-----|------>|                         |
+---------------------------+      |       +-------------------------+
```

Possible Inter-Clouds Interfaces:

- Provisioning
- Signaling
- Control
- Monitoring
- Management

- Transport
- Security
- Naming, Addressing and Translation (if/as needed)

7. Open Security Architecture (OSA): Secure Architecture Models [9]

Open Security Architecture (OSA) provides free frameworks that are easily integrated in applications for the security architecture community. Its patterns are based on schematics that show the information traffic flow for a particular implementation as well as policies implemented at each step for security reasons. The following description of the proposed cloud computing architecture, seen in the figure below, should help the reader envision the components of cloud computing architectures along with descriptions of elements that make it secure. The important entities involved in the data flow are end users, developers, system architects, 3rd party auditors, and the cloud itself.

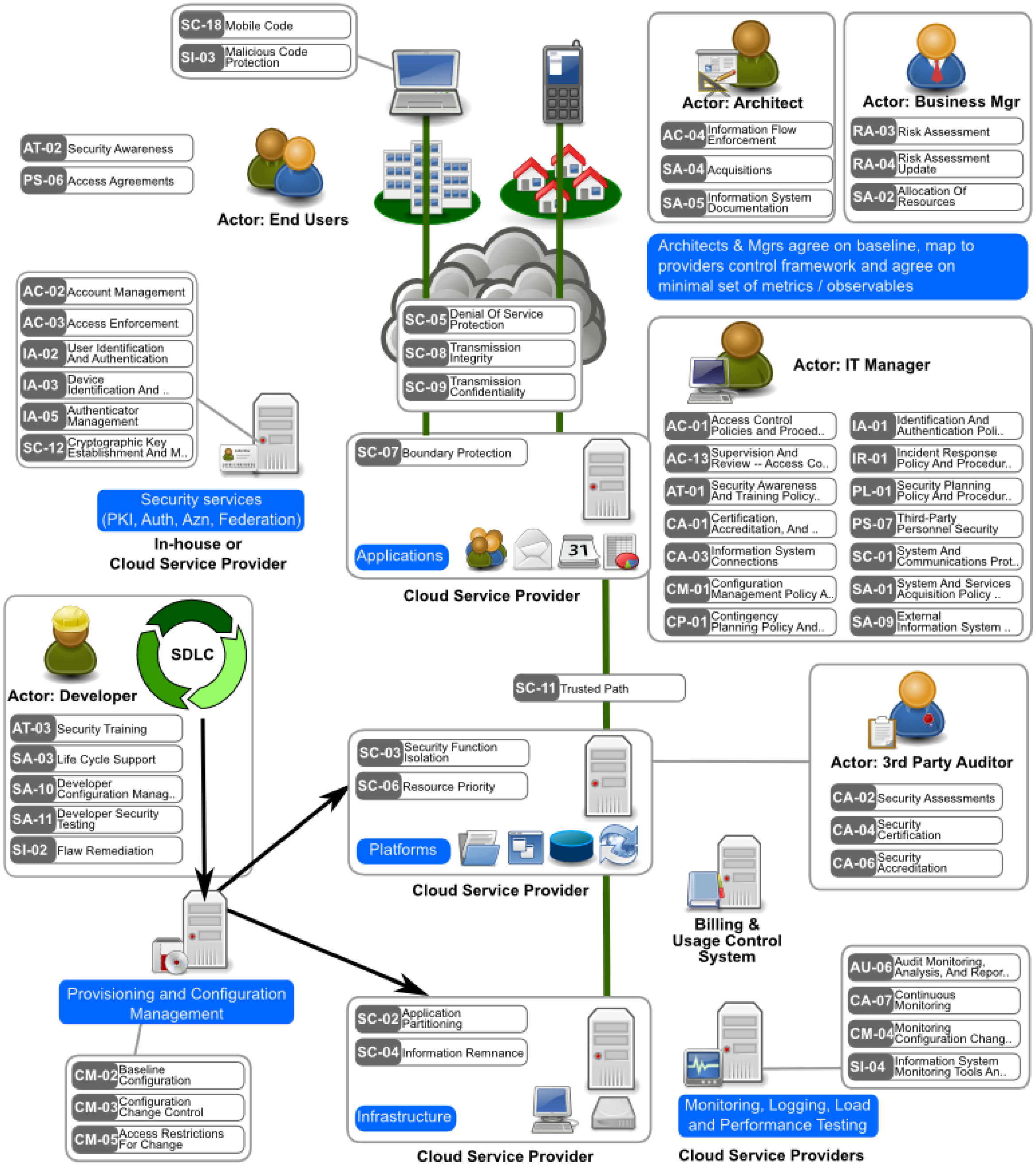
7.1 End Users

End Users need to access certain resources in the cloud and should be aware of access agreements such as acceptable use or conflict of interest. In this model, end user signatures may be used to confirm they are committed to such policies. The client organization should run mechanisms to detect vulnerable code or protocols at entry points such as firewalls, servers, or mobile devices and upload patches on the local systems as soon as they are found. Thus, this approach ensures security responsibilities fall on the end users and on the cloud alike.

However, the cloud needs to be secure from any user with malicious intent that may attempt to gain access to information or shut down a service. For this reason, the cloud should include denial of service (DOS) protection. One way of enforcing DOS protection is done by improving the infrastructure with more bandwidth and better computational power that the cloud has in abundance. However, in the more traditional sense, it involves filtering certain packets that have similar IP source addresses or server requests. The next issue concerning providing cloud services to end users is transmission integrity. One way of implementing integrity is by using secure socket layer (SSL) or transport layer security (TLS) to ensure that sessions are not being altered by a man-in-the-middle attack. At a lower level, the network can be made secure by the use of the secure internet protocol (IPsec). Lastly is transmission confidentiality, or the guarantee that no one is listening in on the conversation between authenticated users and the cloud. The same mechanisms mentioned above can also guarantee confidentiality.

7.2 System Architects

System Architects write the policies that pertain to the installation and configuration of hardware components such as firewalls, servers, routers, and software such as operating systems, thin clients, etc. They designate control protocols to direct the information flow within the cloud such as router update/queuing protocols, proxy server configurations or encrypted tunnels.


SC-18 Mobile Code
SI-03 Malicious Code Protection
AT-02 Security Awareness
PS-06 Access Agreements
Actor: End Users
Actor: Architect
AC-04 Information Flow Enforcement
SA-04 Acquisitions
SA-05 Information System Documentation
Actor: Business Mgr
RA-03 Risk Assessment
RA-04 Risk Assessment Update
SA-02 Allocation Of Resources
Architects & Mgrs agree on baseline, map to providers control framework and agree on minimal set of metrics / observables
AC-02 Account Management
AC-03 Access Enforcement
IA-02 User Identification And Authentication
IA-03 Device Identification And ..
IA-05 Authenticator Management
SC-12 Cryptographic Key Establishment And M..
SC-05 Denial Of Service Protection
SC-08 Transmission Integrity
SC-09 Transmission Confidentiality
Actor: IT Manager
AC-01 Access Control Policies and Proced..
IA-01 Identification And Authentication Poli..
AC-13 Supervision And Review -- Access Co..
IR-01 Incident Response Policy And Procedur..
AT-01 Security Awareness And Training Policy..
PL-01 Security Planning Policy And Procedur..
CA-01 Certification, Accreditation, And ..
PS-07 Third-Party Personnel Security
CA-03 Information System Connections
SC-01 System And Communications Prot..
CM-01 Configuration Management Policy A..
SA-01 System And Services Acquisition Policy ..
CP-01 Contingency Planning Policy And..
SA-09 External Information System ..
SC-07 Boundary Protection
Applications
31
Cloud Service Provider
Security services (PKI, Auth, Azn, Federation)
In-house or Cloud Service Provider
Actor: Developer
SDLC
AT-03 Security Training
SA-03 Life Cycle Support
SA-10 Developer Configuration Manag..
SA-11 Developer Security Testing
SI-02 Flaw Remediation
SC-11 Trusted Path
SC-03 Security Function Isolation
SC-06 Resource Priority
Platforms
Cloud Service Provider
Actor: 3rd Party Auditor
CA-02 Security Assessments
CA-04 Security Certification
CA-06 Security Accreditation
Billing & Usage Control System
Provisioning and Configuration Management
CM-02 Baseline Configuration
CM-03 Configuration Change Control
CM-05 Access Restrictions For Change
SC-02 Application Partitioning
SC-04 Information Remnance
Infrastructure
Cloud Service Provider
AU-06 Audit Monitoring, Analysis, And Repor..
CA-07 Continuous Monitoring
CM-04 Monitoring Configuration Chang..
SI-04 Information System Monitoring Tools An..
Monitoring, Logging, Load and Performance Testing
Cloud Service Providers

08_02_Pattern_011_15_Cloud_Computing.svg
OSA is licensed according to Creative Commons Share-alike.
Please see:http://www.opensecurityarchitecture.org/cms/community/license-terms.

7.3 Developers

Developers building an application in the cloud need to access the infrastructure where the development environment is located. They also need to access some configuration server that allows them to test applications from various views. Cloud computing can improve software development by scaling the software environment through elasticity of resources. For example, one developer can get extra hard space as an on-demand resource instead of placing a work order and waiting for several days to get permission. Developers may desire extra virtual machines for time-consuming processes such as generating test data or performing data analysis. Also, using more processing power from the cloud can help catch up with s development schedule. The cloud also helps developers create multiple evaluation environments for their applications, bypassing the need to incorporate additional security within the application and placing the burden on the cloud provider. One significant drawback of cloud computing at the moment is its limitation to Intel x86 processor architecture. Even if this may very well change in the future, it is another stumbling block that developers and cloud computing experts need to overcome. Software monitoring may be done by monitoring API calls for server requests. With an architectural model where data is centralized, all eyes are focused in one direction, which implies better monitoring, although ultimately the issue rests with the developers/clients on how much effort will be directed in this regard. as Applying security patches is easier with the SaaS approach, as they can be shared with everyone seamlessly rather than finding and patching every machine that has the software installed locally.

7.4 Third Party Auditors

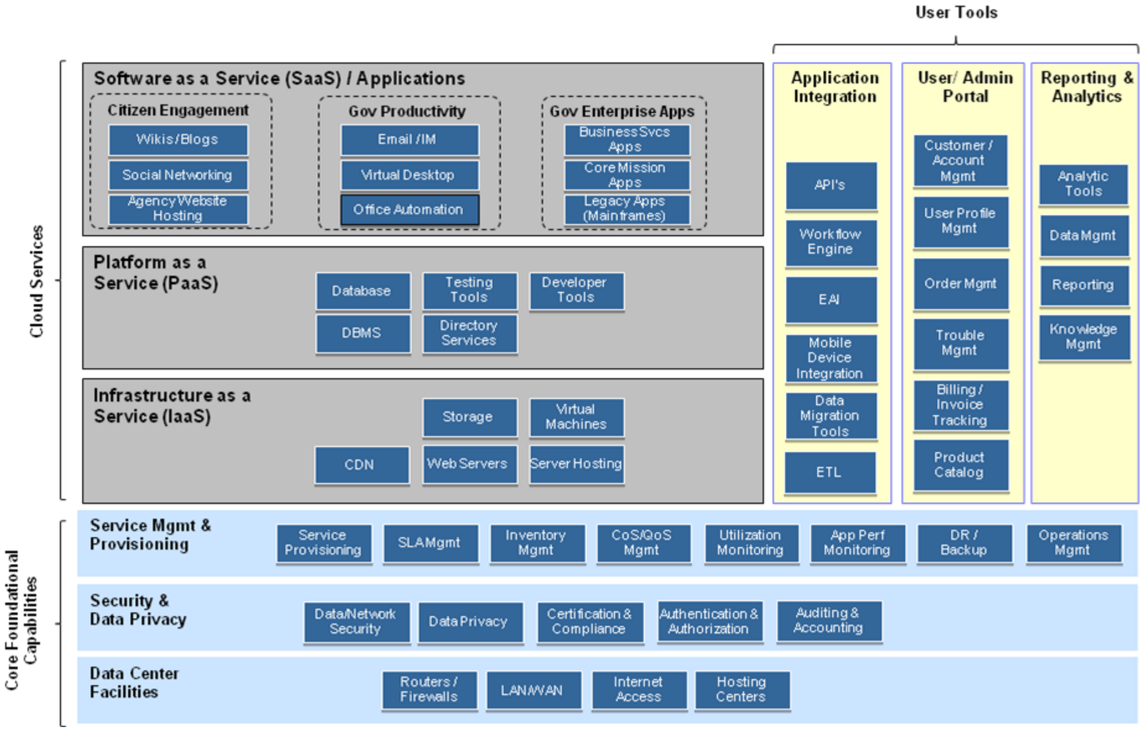
Third Party Auditors are used by clients and providers alike to determine the security of the cloud implementation. Depending on the level of commitment to security and its usefulness in obtaining a competitive edge, a cloud vendor may choose to submit itself to regular security assessments in an attempt to obtain accreditation. The accreditation process needs to be undertaken every three years. In order to lower the constraints on the cloud vendor, some organizations may implement continuous monitoring of the cloud system.

7.5 Overview

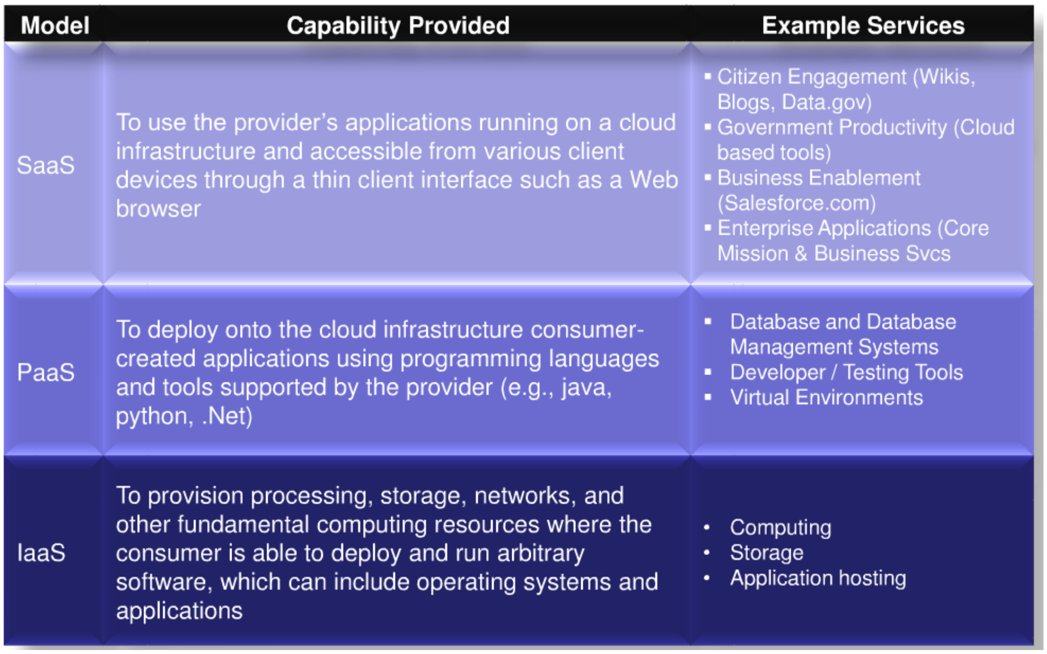
The cloud is the resource that incorporates routers, firewalls, gateways, proxy and storage servers. The interaction among these entities needs to occur in a secure fashion. For this reason the cloud, just like any data center, implements a boundary protection also known as the demilitarized zone (DMZ). The most sensitive information is stored behind the DMZ. Other policies that run in the cloud are resource priority and application partitioning. Resource priority allows processes or hardware requests in a higher priority queue to be serviced first. Application partitioning refers to the usage of one server or storage device for various clients that may have data encrypted differently. The cloud should have policies that divide the users' view of one application from the backend information storage. This may be solved by using virtualization, multiple processors, or network adaptors.

8. GSA: FCCI (Federal Cloud Computing Initiative) [6][7]

8.1 Government Cloud Computing Framework



Delivery Model Overview

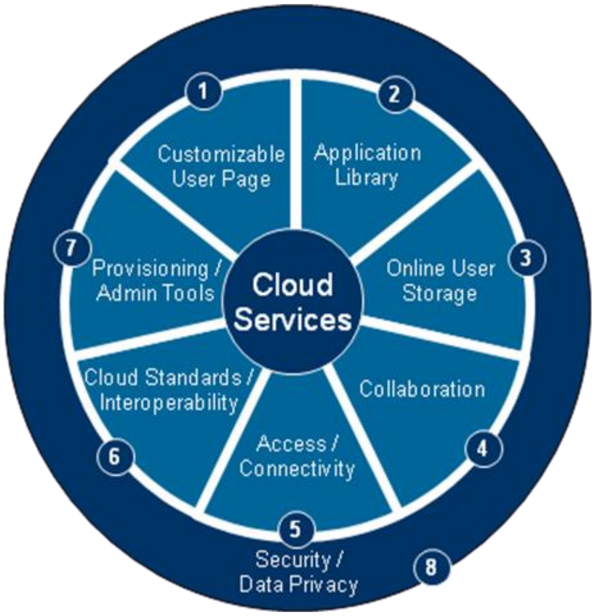


Model	Capability Provided	Example Services
SaaS	To use the provider's applications running on a cloud infrastructure and accessible from various client devices through a thin client interface such as a Web browser	▪ Citizen Engagement (Wikis, Blogs, Data.gov) ▪ Government Productivity (Cloud based tools) ▪ Business Enablement (Salesforce.com) ▪ Enterprise Applications (Core Mission & Business Svcs
PaaS	To deploy onto the cloud infrastructure consumer-created applications using programming languages and tools supported by the provider (e.g., java, python, .Net)	▪ Database and Database Management Systems ▪ Developer / Testing Tools ▪ Virtual Environments
IaaS	To provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications	• Computing • Storage • Application hosting

Federal Government Considerations for Cloud Computing:

- Security & Data Privacy: Offer different levels of security and data privacy based on the application and nature of the services provided. Potential to standardize on Low, Med and High categories for simplicity.
- Delivery & Operations: Enable adoption of Cloud Computing services in different Cloud models including Public, Private, Hybrid, and Community models.
- Interoperability & Integration: Develop interoperability standards in conjunction with industry to provide interoperability at the data infrastructure, platform, and application levels.

8.1.1 Federal Cloud Computing Cloud Program Services Model

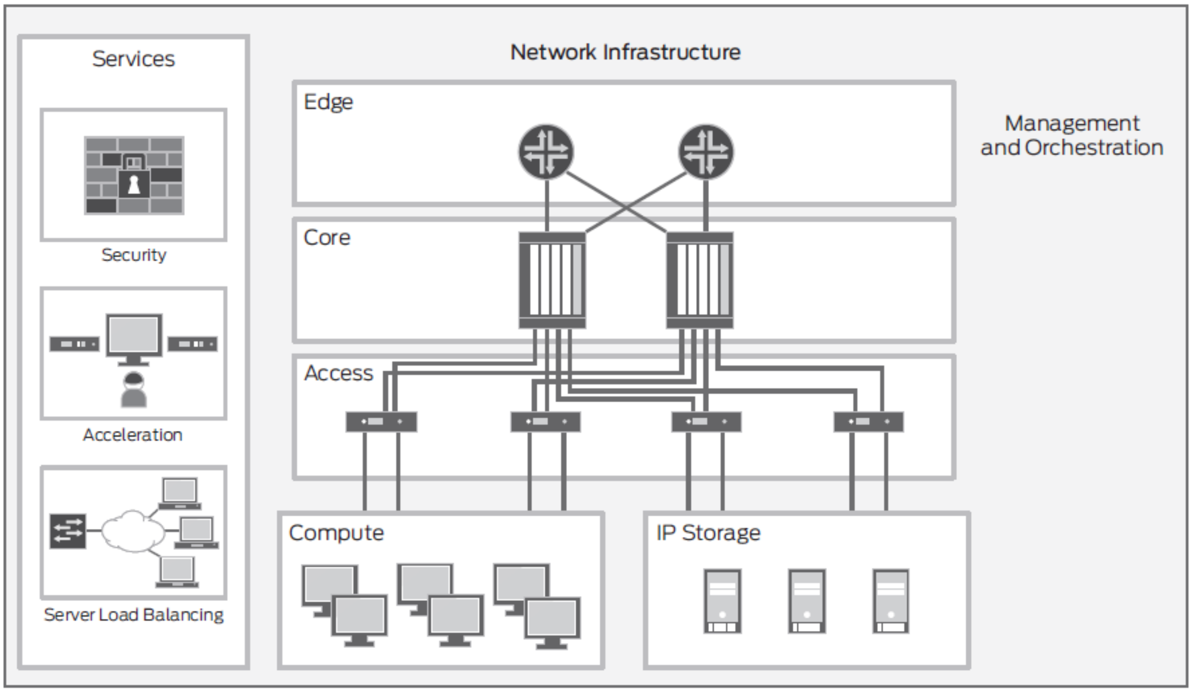


Components	Description
Customizable User Page	• One stop shop/ single view for users to manage interactions with Cloud Services.
Application Library	• Applications and content made available to users through a simple GUI. These applications / content can be downloaded easily from the Library.
Online User Storage	• Online storage for users to maintain and manage individual user files, data and objects.
Collaboration	• Widgets that allow users to collaborate and generate content. These widgets may include Wikis, Blogs, and IM.
Access /	• Network connectivity and devices to access Cloud Services.

Connectivity	
Cloud Standards/ Interoperability	• Cloud standards allowing the integration and interoperability of services from multiple Clouds.
Provisioning / Admin Tools	• Provisioning and administrative tools to provide control of user profiles, access technical support, and manage privileges and authorization to applications and content.
Security /Data Privacy	• Applying security framework and data privacy standards for Federal Cloud Computing.

9. Juniper Network: Cloud-ready Data Center Reference Architecture

Data centers run the applications that deliver business processes and services and have been essential corporate assets that connect all servers, applications, and storage services. The cloud-computing paradigm can be applied to data center network designs to meet a variety of business and application requirements. In [23], Juniper Network introduces its architectural model and its offerings in support of data center and cloud computing networks. The purpose of this reference architecture is to communicate Juniper's conceptual framework and architectural philosophy in creating data center and cloud computing networks for their customers.



The above figure illustrates the framework that Juniper employs to envision the data center network at its highest level. It includes the following areas and their functional interrelationships:

- Network Infrastructure: provides connectivity and transport for applications and services between users and the data center, within the data center, and across multiple data centers. The Network infrastructure has three main sub-components, namely the access network, the core network, and the edge network.
    - Access network: provides connectivity to all shared enterprise servers, applications, devices.
    - Core network: provides a fabric for high-speed packet switching between multiple access network devices.
    - Edge network: provides the communication links to end user networks of various types.
- Compute and Storage: represents the compute and storage infrastructure appropriate for applications (rack-mount and chassis-based, cost-effective and multi-core, with unstructured content and highly structured transaction databases). The compute and storage functional area hosts all business applications such as Enterprise Resource Planning (ERP), SaaS, SOA and Web 2.0 applications (among others).
- Services: supports applications with security, user verification, and entitlement and application support, including application acceleration, deep packet inspection (DPI), and load balancing.
- Management and Orchestration: ties together all of the elements of the cloud-computing infrastructure, enabling efficient and responsive monitoring, management, and planning.

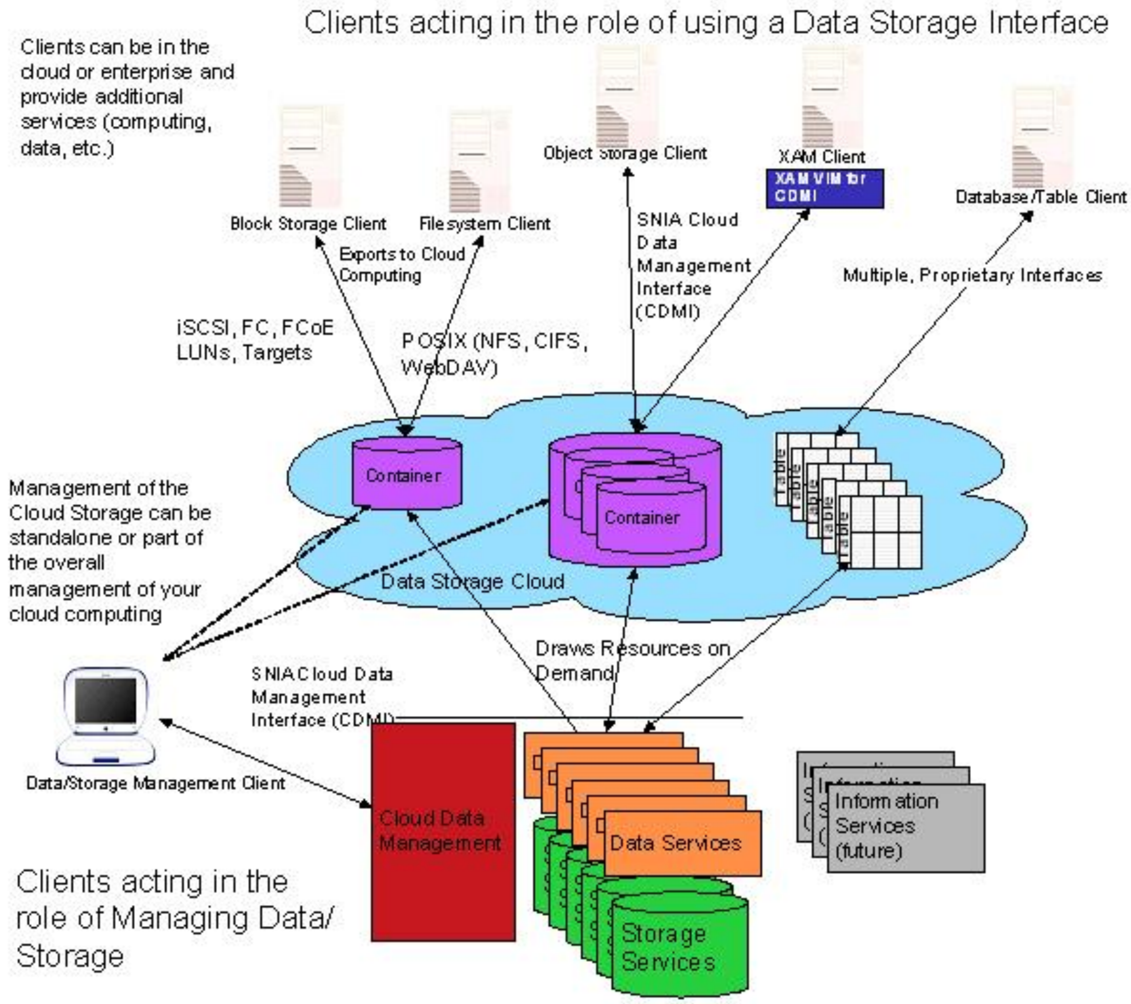
10. SNIA standard: Cloud Data Management Interface [3]

The Storage Networking Industry Association (SNIA) announced at SNW Spring 2010 the formal approval of the Cloud Data Management Interface (CDMI) as a SNIA Architecture Standard. This milestone marks the first industry-developed open standard for cloud computing and will allow for interoperable cloud storage implementations from cloud service providers and storage vendors.

SNIA proposes a formal term for cloud storage, i.e. Data Storage as a Service (DaaS), defined as "delivery over a network of appropriately configured virtual storage and related data services, based on a request for a given service level."[1] SNIA defines CDMI, the functional interface that applications may use to create, retrieve, update, and delete data elements from the cloud.

The Reference Model for Cloud Storage Interfaces is shown below:

[1] ***Note:*** *Cloud Security Alliance regards "Storage as a Service" as a specific sub-offering within the IaaS 'family'.*


Clients acting in the role of using a Data Storage Interface
Clients can be in the cloud or enterprise and provide additional services (computing, data, etc.)
Object Storage Client
XAM Client
XAM VIM for CDMI
Database/Table Client
Block Storage Client
Filesystem Client
Exports to Cloud Computing
SNIA Cloud Data Management Interface (CDMI)
Multiple, Proprietary Interfaces
iSCSI, FC, FCoE LUNs, Targets
POSIX (NFS, CIFS, WebDAV)
Container
Container
Table
Management of the Cloud Storage can be standalone or part of the overall management of your cloud computing
Data Storage Cloud
Draws Resources on Demand
SNIA Cloud Data Management Interface (CDMI)
Data/Storage Management Client
Cloud Data Management
Data Services
Information Services (future)
Storage Services
Clients acting in the role of Managing Data/ Storage

This model shows multiple types of cloud data storage interfaces that are able to support both legacy and new applications. All of the interfaces allow storage to be provided on demand, drawn from a pool of resources. Capacity is drawn from a pool of storage capacity provided by storage services. The data services are applied to individual data elements, as determined by the data system metadata. Metadata specify data requirements on the basis of individual data elements or on groups of data elements (containers).

10.1 Cloud Data Management Interface (CDMI):

As shown in the above figure “Cloud Storage Reference Model”, the SNIA Cloud Data Management Interface (CDMI) is the functional interface that applications may use to create, retrieve, update, and delete data elements from the cloud. As part of this interface, the client will be able to discover the capabilities of the cloud storage offering and to use this interface to manage containers and the data that are placed in them. In addition, data system metadata can be set on containers and their contained data elements through this interface.

This interface may also be used by administrative and management applications to manage containers, domains, security access, and monitoring/billing information, even for storage that is functionally accessible by legacy or proprietary protocols. The capabilities of the underlying storage and data services are exposed so that clients can understand the offering.

10.2 Data Storage Interfaces:

10.2.1 Existing Data Storage Interfaces

An important part of any DaaS offering is the support of legacy clients. Support is accommodated with existing standard protocols such as iSCSI (and others) for block storage and CIFS/NFS or WebDAV for file network storage. In the case of block storage, a LUN, or virtual volume, is the granularity of allocation. For file protocols, a file system is the unit of granularity. In either case, the actual storage space can be thin provisioned and billed for based on actual usage. Data services, such as compression and deduplication, can be used to further reduce the actual space consumed.

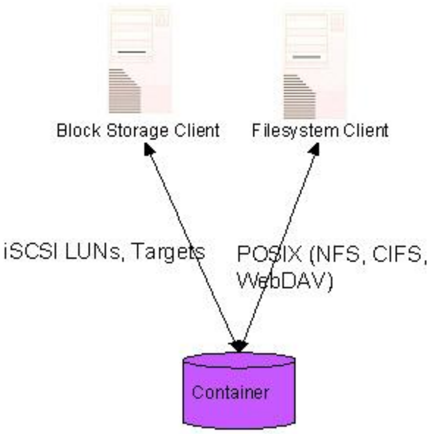


Fig. Existing Data Storage Interface Standards

In this model, we abstract the underlying storage space exposed by these interfaces using the notion of a container. A container is not only a useful abstraction for storage space, but also serves as a grouping of the data stored in it and a point of control for applying data services in the aggregate.

Managing this storage is typically done out-of-band through these standard data storage interfaces, either through an API or, more commonly, through an administrative browser-based user interface. This interface may be used to invoke other data services as well, such as snapshots and cloning.

10.2.2 Storage Interfaces for Database/Table Data

Another type of DaaS offering is one of simple table space storage, allowing for horizontal scaling of the database-like operations that certain applications need. Rather than virtualizing relational database instances, table space storage offers a new data storage interface of limited functionality with the emphasis on scalability rather than features. Scalability allows the tables to be partitioned across multiple nodes based on common key values, affording horizontal scalability at the expense of functions that can typically only be implemented by a vertically-scaled relational database.

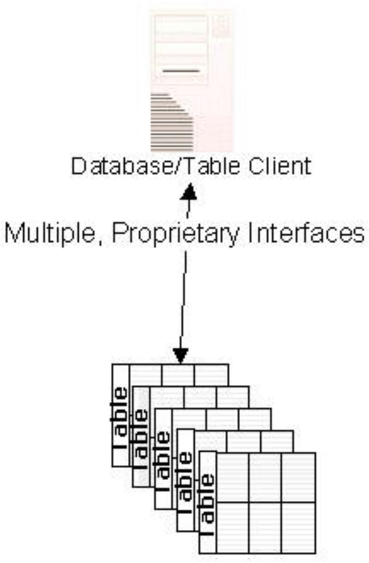


Fig. Storage Interfaces for Database/Table Data

10.2.3 Storage Interfaces for Object Storage Client Data

This type of interface treats every data object as accessible via a unique URL. It can then be fetched using the standard HTTP protocol and a browser can be used to invoke the appropriate application to deal with the data.

Each data object is created, retrieved, updated, and deleted (CRUD semantics) as a separate resource. In this type of interface a container, if used, is a simple grouping of data objects for convenience. Nothing prevents the concept of containers from being hierarchical, although any given implementation might support only a single level of such.

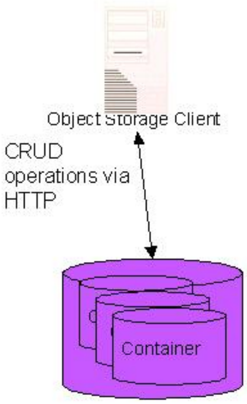


Fig. Storage Interfaces for Object Storage Client Data

10.3 Data Management in the Cloud

To address the needs of enterprise applications with cloud storage, there is increasing pressure to offer better quality of service and the deployment of additional data services. However, cloud storage may lose its benefit of simplicity and abstraction of complexity as additional data services are applied and the need to manage these services increases. One can hardly have cloud storage customers setting up backup schedules through dedicated user interfaces, deploying data services individually for their data elements, and so on.

The SNIA Storage Industry Resource Domain Model (SIRDM) gives us a way to minimize this complexity and address the need for cloud storage to remain simple. By using the different types of metadata discussed in the model for a cloud storage interface, we can create an interface that allows offerings to meet the requirements of the data without adding complexity to the management of that data.

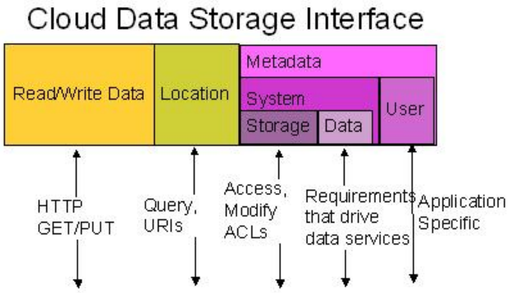
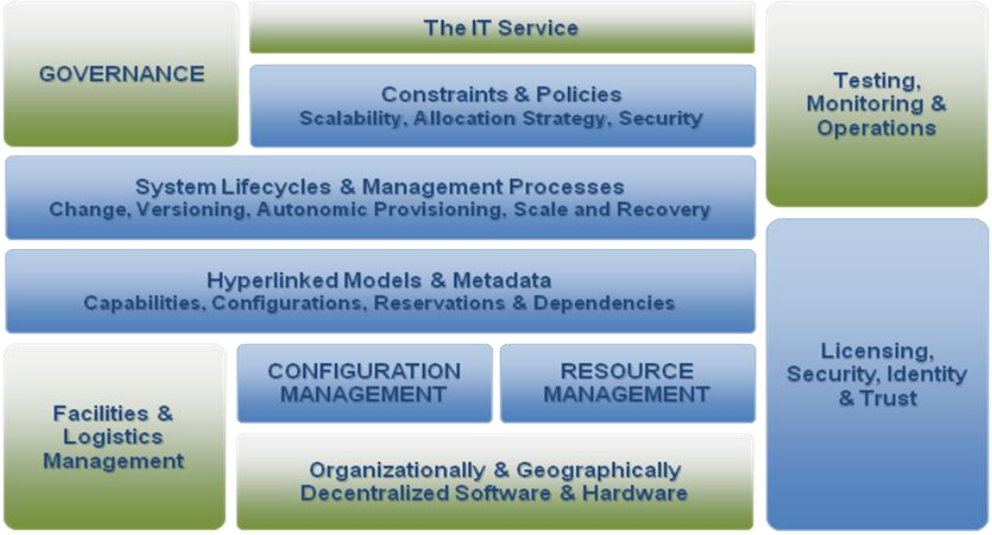


Fig. Using the Resource Domain Model

By supporting metadata in a cloud storage interface standard and prescribing how the storage system and data system metadata are interpreted to meet the requirements of the data, we can retain the simplicity required by the cloud storage paradigm and still address the requirements of enterprise applications and their data.

11. Elastra: A Cloud Technology Reference Model for Enterprise Clouds [22]


GOVERNANCE
The IT Service
Constraints & Policies
Scalability, Allocation Strategy, Security
Testing, Monitoring & Operations
System Lifecycles & Management Processes
Change, Versioning, Autonomic Provisioning, Scale and Recovery
Hyperlinked Models & Metadata
Capabilities, Configurations, Reservations & Dependencies
Facilities & Logistics Management
CONFIGURATION MANAGEMENT
RESOURCE MANAGEMENT
Organizationally & Geographically Decentralized Software & Hardware
Licensing, Security, Identity & Trust

- Facilities & Logistics Management, Organizationally & Geographically Decentralized Software & Hardware: The bbasic data center, which is now global and possibly cross-organizational and exposes power and cooling information.
- Licensing, Security, Identity & Trust: the ccontrol point for compliance and auditing, which adds trust, identity, and licensing.
- Configuration management: deals with HW/SW/network/storage settings, Ssoftware packages, and dependencies
- Resource management: deals with reservations from a pool of excess capacity in storage, computing, and network.
- Hyperlinked Models & Metadata: what uses or contains what other things.
- System Lifecycles & Management Processes: when and how can things change.
- Governance: determining who the has authority or responsibility to make changes, and how those changes are made.
- Constraints & Policies: describes how concerns are addressed in the design.
- Testing, Monitoring & Operations: Describes how changes are managed and verified.

12. Comparison

We briefly compare all the presented models on different abstraction levels (business/architecture), and indicate for each model the key difference in the following table:

Model	Comparison on Business level	Comparison on Architecture level	Key Difference
DMTF	For general CC systems	Defines actors (cc provider/user/ develop), interfaces for interaction between actors, data artifacts for information exchanged via interfaces, and profiles for contexts.	Focuses on provider interfaces, i.e. how to support interaction among actors.
Use case group	For general CC systems	Defines actors (cc provider/user/ develop), cloud components, service models, diff interfaces for diff service models, and service management.	Compared to DMTF, adds service models/cloud components/ service management.
IBM	For general CC systems	Defines actors (cc provider/user/ develop), cloud components, service models, business/operational support services for cloud providers.	Compared to use case group's model, adds more details for service management.
CSA	For general CC systems	Defines resources stack, IaaS/PaaS/SaaS, and their relationship /dependency.	The only one that defines the dependency and build up of IaaS to PaaS to SaaS.
Cisco	For general CC systems	Defines a layered model which considers resources, security, service, and consumers layer.	The only layered architecture.
IETF	For general CC systems	Defines both intra-cloud and inter-cloud reference frameworks. The intra-cloud framework defines the layers to support the general requirements of cloud services.	The only one considering an inter-cloud framework.
OSA	Specifically for	Defines actors (users/system	Focuses on security

	cloud security	architects/developers/3rd party auditors), security policies at each step, following information traffic flow.	architecture.
GSA	Specifically for government cloud services	Defines government service models, cloud components, user tools, federal cloud services model.	Focuses on government's unique needs, tailoring all components to those needs.
Juniper	Specifically for data center network solution	Defines Juniper's conceptual framework and architectural philosophy in creating data center and cloud computing networks.	Focuses on data center and cloud computing networks.
SNIA	Specifically for cloud storage	Defines "DaaS" and CDMI (the interface for accessing and managing cloud storage).	1st industry-developed open standard for cloud computing. Focuses on cloud storage.
Elastra	Specifically for enterprise cloud	Defines a management framework for enterprise clouds that focuses on management of different components and lifecycles.	Compared to IBM, Elastra considers only cloud management.

13. Reference

[1] Cloud Computing Use Cases White Paper, http://groups.google.com/group/cloud-computing-use-cases

[2] DMTF, "Interoperable Clouds White Paper", http://www.dmtf.org/about/cloud-incubator/DSP_IS0101_1.0.0.pdf

[3] SNIA: http://cdmi.sniacloud.com/

[4] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus In Cloud Computing V2.1", www.cloudsecurityalliance.org/csaguide.pdf

[5] SNIA, "Cloud Storage for Cloud Computing", www.snia.org/cloud/CloudStorageForCloudComputing.pdf

[6] GSA, "Cloud Computing Initiative", http://www.info.apps.gov/sites/default/files/Cloud_Computing_Initiative_Briefing_Book_0.doc

[7] GSA, "Cloud Computing Initiative Vision and Strategy Document (DRAFT)", http://info.apps.gov/sites/default/files/Cloud_Computing_Strategy_0.ppt

[8] Department of Homeland Security, **"**Cloud Computing from the Security Perspective**",** http://www.info.apps.gov/sites/default/files/Cloud_Computing_Security_Perspective.doc

[9] Open Security Architecture (OSA), "Cloud Computing Patterns", http://www.opensecurityarchitecture.org/cms/library/patternlandscape/251-pattern-cloud-computing

[10] Traian Andrei, "Cloud Computing Challenges and Related Security Issues", http://www.cse.wustl.edu/~jain/cse571-09/ftp/cloud/index.html

[11] Jinesh Varia (@Amazon Web Services), "Cloud Architectures", http://jineshvaria.s3.amazonaws.com/public/cloudarchitectures-varia.pdf

[12] Gerald Kaefer (@Siemens), "Cloud Computing Architecture", http://www.sei.cmu.edu/library/assets/presentations/Cloud%20Computing%20Architecture%20-%20Gerald%20Kaefer.pdf

[13] "A Reference Architecture for Cloud Lifecycle Management", http://www.spokenword.org/program/1278676

[14] IBM, “Cloud Computing”, http://download.boulder.ibm.com/ibmdl/pub/software/dw/wes/hipods/Cloud_computing_wp_final_8Oct.pdf

[15] IBM, “Private Cloud Computing Architectural Concepts”, http://www.google.com/url?sa=t&source=web&cd=1&ved=0CCAQFjAA&url=http%3A%2F%2Fwww-05.ibm.com%2Ffr%2Fevents%2Fst_cloud_computing%2FCloud_Architecture_Overview_STCC16032010.pdf&ei=tgrLTMXxEIPGlQec-4yuAQ&usg=AFQjCNHAAILq03xIZxq67J6wOHlthr-2sg

[16] IBM, “IBM Smart Analytics Cloud”, www.redbooks.ibm.com/redbooks/pdfs/sg247873.pdf

[17] IBM, “Cloud Computing: Save Time, Money, and Resources with a Private Test Cloud”, www.redbooks.ibm.com/redpapers/pdfs/redp4553.pdf.

[18] IBM, "Cloud Security Guidance - IBM Recommendations for the Implementation of Cloud Security", http://www.redbooks.ibm.com/redpapers/pdfs/redp4614.pdf

[19] IBM, “IBM Cloud Computing – Service Management”, http://www.google.com/url?sa=t&source=web&cd=1&ved=0CB8QFjAA&url=https%3A%2F%2Fqp.research.ibm.com%2FLotusQuickr%2Fsldemos%2FMain.nsf%2F0%2F6DFC3C4967F62E188525762C006F21DC%2F%24file%2FLindquist_IBM_Cloud_Computing-Service_Management_v4_D.pdf&ei=4BDLTLGvGIGglAfKzrnxAQ&usg=AFQjCNFE6ANMY9VBtnfX66FGR8sF2lzNhA

[20] IBM, “Cloud Computing - The Importance of Integrated Service Management”, http://www.google.com/url?sa=t&source=web&cd=1&ved=0CBMQFjAA&url=https%3A%2F%2Fwww-950.ibm.com%2Fevents%2Fwwe%2Fgrp%2Fgrp004.nsf%2FvLookupPDFs%2FCloud%2520Computing%2520-%2520The%2520Importance%2520of%2520Integrated%2520Service%2520Management%2520Minneapolis%25202010%2F%24file%2FCloud%2520Computing%2520-%2520The%2520Importance%2520of%2520Integrated%2520Service%2520Management%2520Minneapolis%25202010.pdf&ei=0B_LTPqyLIH_8AbD_rWrAQ&usg=AFQjCNH_CYyQgidqxo7KwGHbtd4zWqrE7w

[21] CISCO, “Cisco Cloud Computing - Data Center Strategy, Architecture, and Solutions”, http://www.google.com/url?sa=t&source=web&cd=1&ved=0CB0QFjAA&url=http%3A%2F%2Fwww.cisco.com%2Fweb%2Fstrategy%2Fdocs%2Fgov%2FCiscoCloudComputing_WP.pdf&ei=Mx_LTOHwOIL68AaPk_zFAQ&usg=AFQjCNEgDF69wa7aswI7k7gmINpy6QhfHg

[22] Stuart Charlton, “Cloud Computing and the Next Generation of Enterprise Architecture”, http://www.slideshare.net/StuC/cloud-computing-and-the-nextgeneration-of-enterprise-architecture-cloud-computing-expo-2008-presentation

[23] Juniper Networks, “Cloud-ready Data Center Reference Architecture”, www.juniper.net/us/en/local/pdf/reference-architectures/8030001-en.pdf

[24] IETF internet-draft, “Cloud Reference Framework”, http://www.ietf.org/id/draft-khasnabish-cloud-reference-framework-00.txt